<u>In the Claims:</u>

Please cancel claims 11, 12, 33-37, and 47-57 without prejudice or disclaimer.

Please amend claims 1, 3-5, 9, 13-14, 16-19, and 38 as follows:

1. (CURRENTLY AMENDED) A computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory, and <u>a</u> nonvolatile storage, the computerized game controller being operable to control a computerized wagering game;

an operating system comprising[[:]]<u>an operating system kernel that executes a system handler application,</u> [[a]] <u>the</u> system handler application operable to dynamically link with ~~at least one~~ <u>a plurality of</u> gaming program <u>shared</u> object<u>s and load said gaming program shared objects</u>;

<u>the system handler application</u> having an Application Program Interface having functions callable from the gaming program <u>shared</u> object<u>, the Application Program Interface having a plurality of gaming functions callable by and used by at least some of the shared objects,</u>

<u>the system handler application operable to initiate a game based on data variables stored in the nonvolatile storage, the system handler application operable to write game data variables to at least one of the game state storage and nonvolatile storage</u> ~~and~~

~~an operating system kernel that executes the system handler application.~~

2. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises an event handler.

- 3 -

3. (CURRENTLY AMENDED) The computerized wagering game apparatus of claim 1, wherein the system handler application comprises software having the ability when executed to:

unload a previous gaming program <u>shared</u> object if a previous object has been loaded;

load a new gaming program <u>shared</u> object; and

execute the new gaming program <u>shared</u> object.

4. (CURRENTLY AMENDED) The computerized wagering game apparatus of claim 1, wherein data variables modified by the gaming program <u>shared</u> objects are stored <u>by the system handler</u> application in <u>the</u> nonvolatile storage <u>and a game state storage</u>, and <u>the system handler application</u> functions <u>to</u> verify that the operating system or code for a shared object has not changed.

5. (CURRENTLY AMENDED) The computerized wagering game apparatus of claim 4, further comprising a <u>wherein the</u> game state <u>storage</u> ~~device~~ providing a variable name index to associated variable data locations within the nonvolatile storage.

6. (ORIGINAL) The computerized wagering game apparatus of claim 4, wherein changing a data variable in nonvolatile storage causes execution of a corresponding callback function in <u>one of</u> the ~~system handler application~~ gaming program <u>shared</u> object<u>s of the system handler application</u>.

7. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the computerized game controller comprises an IBM PC-compatible computer.

8. (ORIGINAL) The computerized wagering game apparatus of claim 1, wherein the operating system kernel is a Linux operating system kernel.

- 4 -

9. (CURRENTLY AMENDED) The computerized wagering game apparatus of claim 8, wherein the shared objects include a plurality of device handlers and the Linux operating system kernel has at least one selected device handler disabled.

10. (ORIGINAL) The computerized wagering game apparatus of claim 9, wherein the at least one selected device handler that is disabled is selected from the group consisting of a keyboard handler, an I/O port handler, a network interface handler, a storage device controller handler, and a I/O device handler.

11. (CANCELLED)

12. (CANCELLED)

13. (CURRENTLY AMENDED)   A method of managing data in a computerized wagering game apparatus via a system handler application, comprising:

executing an operating system which then loads and operates the system handler application, the system handler application operable to dynamically link with a plurality of gaming program shared objects and load said shared objects, the system handler application having an Application Program Interface having a plurality of functions callable from at least some of the shared objects, the system handler application operable to initiate a game based on data variables stored in a nonvolatile storage and the system handler application operable to write game data variables to the nonvolatile storage,

the system handler application then loading a first ~~program~~ shared object and providing an Application Program Interface ~~having~~ function[s] called by the first ~~program~~ shared object,

the system handler application then executing the first ~~program~~ shared object,

- 5 -

the system handler application then storing data variables in the nonvolatile storage, such that a second ~~program~~ shared object later loaded can access the data variables in nonvolatile storage,

the application handler application then unloading the first ~~program~~ shared object, and

the system handler application then loading a second ~~program~~ shared object.

14. (CURRENTLY AMENDED) The method of claim 13, further comprising the system handler application executing a corresponding callback function upon alteration of variable data in nonvolatile storage.

15. (ORIGINAL) The method of claim 13, further comprising handling events via the system handler application.

16. (CURRENTLY AMENDED) A secure computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory, and a nonvolatile storage, the computerized game controller being operable to control a computerized wagering game;

an operating system comprising an operating system kernel that executes a system handler application, the system handler application operable to dynamically link with a plurality of gaming program shared objects and load said shared objects, said shared objects including a plurality of device handlers;

the system handler application having an Application Program Interface having functions callable from and used by at least some of the shared objects,

the system handler application operable to initiate a game based on data variables stored in the nonvolatile storage, the system handler application operable to write game data variables to the nonvolatile storage,

- 6 -

the operating system controlled by a general-purpose computer[[,]] and comprising an the operating system kernel that is being customized to disable selected device handlers, and a program that

the system handler application further loads a first program game shared object and the first program shared game object calls up a gaming function from within an Application Program Interface.

17. (CURRENTLY AMENDED) A secure computerized wagering game apparatus, comprising:

a computerized game controller having a processor, memory, and a nonvolatile storage, the computerized game controller being operable to control a computerized wagering game;

an operating system comprising an operating system kernel that executes a system handler application, the system handler application operable to dynamically link with a plurality of gaming program shared objects and load said shared objects;

the system handler application having an Application Program Interface having functions callable from at least one of the shared objects, the system handler application operable to initiate a game based on data variables stored in the nonvolatile storage, the system handler application operable to write game data variables to the nonvolatile storage,

the operating system controlled by a general-purpose computer comprising and the nonvolatile storage that stores program variables, such that loss of power does not result in loss of the state of the computerized wagering game system, and a program the system handler application that loads a first program game shared object and the first program game shared object calls up a gaming function from within an Application Program Interface.

18. (CURRENTLY AMENDED) The secure computerized wagering game system of claim 17, further comprising at least one gaming program object, such that a wherein the system application handler loads and executes a single gaming program shared object is

- 7 -

~~loaded and executed~~ at any one time but <u>other</u> ~~gaming program~~ <u>shared</u> objects are operable to share data via the program variables <u>stored</u> in nonvolatile storage.

19.   (CURRENTLY AMENDED) A machine-readable medium with instructions thereon, the medium being within a wagering apparatus, the instructions when executed operable to cause a computer to:

cause a system handler application to load and execute <u>a plurality of</u> gaming program <u>shared</u> objects<u>, one at a time</u>;

cause a loaded gaming program <u>shared</u> object to call up a library of functions provided by <u>an Application Program Interface of</u> the system handler application;

load<u>ing</u> a first <u>share object,</u> ~~program, object from the library,~~

~~execute~~ <u>executing</u> the first ~~program~~ <u>shared</u> object, including having the first ~~program~~ <u>shared</u> object call up a function from within ~~an~~ <u>the library of functions of the</u> Application Program Interface, <u>the executing of the first shared object generating data variables,</u>

~~store~~ <u>storing</u> data variables in nonvolatile storage, such that a second ~~program~~ <u>shared</u> object ~~in the library~~ later loaded <u>and executed</u> can access the data variables in nonvolatile storage,

unload the first ~~program~~ <u>shared</u> object, and

load the second ~~program~~ <u>shared</u> object, including having the second ~~program~~ <u>shared</u> object call up a function from within ~~an~~ <u>the library of</u> Application Program Interface.

20. (ORIGINAL) The machine-readable medium of claim 19, with further instructions operable when executed to cause a computer to execute a corresponding callback function upon alteration of variable data in the nonvolatile storage.

21. (ORIGINAL) The machine-readable medium of claim 19, with further instructions operable when executed to cause a computer to manage events via the system handler application.

- 8 -

22. (CANCELLED)

23. (CANCELLED)

24. (CANCELLED)

25. (CANCELLED)

26. (CANCELLED)

27. (CANCELLED)

28. (CANCELLED)

29. (CANCELLED)

30. (CANCELLED)

31. (CANCELLED)

32. (CANCELLED)

33. (CANCELLED)

34. (CANCELLED)

35. (CANCELLED)

36. (CANCELLED)

37. (CANCELLED)

38. (CURRENTLY AMENDED) The apparatus of claim 1, wherein the system handler application and kernel work in communication to hash [the] system handler code and the operating system kernel code.

39. (CANCELLED)

40. (CANCELLED)

41. (CANCELLED)

42. (CANCELLED)

43. (CANCELLED)

44. (CANCELLED)

45. (CANCELLED)

46. (CANCELLED)

47. (CANCELLED)

48. (CANCELLED)

49. (CANCELLED)

50. (CANCELLED)

51. (CANCELLED)

52. (CANCELLED)

53. (CANCELLED)

54. (CANCELLED)

55. (CANCELLED)

56. (CANCELLED)

57. (CANCELLED)